2014

Towards proper-inconsistency in weldability prediction using k-nearest neighbor regression and generalized regression neural network with mean acceptable error

Junheung Park
Wayne State College

Kyoung-Yun Kim
Wayne State University, kykim@eng.wayne.edu

Raj Sohmshetty
Ford Motor Company

Towards Proper-Inconsistency in Weldability Prediction Using k-Nearest Neighbor Regression and Generalized Regression Neural Network with Mean Acceptable Error

Junheung Park[1], Kyoung-Yun Kim[1*], and Raj Sohmshetty[2]

[1]Department of Industrial and Systems Engineering
Wayne State University
Detroit, MI 48202, USA

[2]Ford Motor Company
Dearborn, MI 48124, USA

ABSTRACT

A significant inconsistency problem exists in the quality of resistance spot welding, and yet it offers various advantages in production. These inconsistent welding data can be eliminated using anomaly detection or instance selection methods. However, in the weldability prediction problem, this inconsistency we refer to as proper-inconsistency, may not be eliminated since it can be used to extract additional information. In this research, we examine the effects of this inconsistency on prediction performance using two machine learning methods, k-Nearest Neighbors (kNN) regression and Generalized Regression Neural Network, in order to identify an approach towards tackling the proper-inconsistency problem in weldability prediction. We also propose a new prediction performance measure, Mean Acceptable Error (MACE), for prediction models in the presence of proper-inconsistency. The proposed method is tested with actual weldability test data.

1. INTRODUCTION

In this study, we investigate the inconsistency problem that frequently occurs during the resistance spot welding (RSW) process. RSW is one of the most widely used metal joining processes in many industries, including the automobile industry [8]. A significant inconsistency problem still exists in the quality of welding, and yet it offers various advantages, such as high speed, high volume operations, and high rate production [1]. Many researchers report that some of the important factors that determine the welding quality include weld current, time, force, electrode displacement, temperature variation, and dynamic resistance [1, 8].

The reliability of RSW is one of the main factors that affects production costs. Several studies have been conducted to predict the weldability of the different welding parameters in order to support various tasks, such as quality monitoring and material selection [9, 11]. Due to the complex nature of the welding process, the data collected to construct a prediction model often contains a significant amount of inconsistency, especially where data instances that correspond to the same welding parameters have different welding quality measures (such as nugget width).

In general, this type of inconsistent data is treated as noise in machine learning literature. If data contains noise, the model built from this data is likely to be unreliable. Therefore, the inconsistent data are often eliminated using anomaly detection or instance selection methods. However, in the weldability prediction problem, the inconsistency, which we call *proper-inconsistency*, may not be eliminated since the inconsistency can be used to extract additional important information. For instance, the electrode wears off as it undergoes the welding process. It may lead to deteriorated welding quality. However, the information about the electrode wear is usually not included in the physical test data. If the data is recorded as in the order of the actual physical experiments, one can perform a quality monitoring task by identifying where and when the inconsistency takes place.

In the presence of *proper-inconsistency*, it is inappropriate to perform the same approach generally employed with machine learning algorithms. Most of these machine learning algorithms aim to minimize the overall errors for all data instances (i.e., errors between prediction and target values). Due to the numerical characteristics of *proper-inconsistency,* it is likely to achieve vague prediction results from the model. For instance, suppose there are two inconsistent data sets. They both have the same welding parameters that are used as inputs to construct the machine learning model. As an illustration, if their nugget width values are 5mm and 0mm, respectively, the prediction model will likely end up with a nugget width of about 2.5mm. This result has the minimum error for both data instances. However, in the case of *proper-inconsistency*, it might be more beneficial to predict these data close to

* Corresponding author: Tel.: (313) 577-4396; E-mail: kykim@eng.wayne.edu

one of its target values, either 5mm or 0mm, instead of 2.5mm. In order to perform such predictions, we consider the k-Nearest Neighbor (kNN) regression with a small number of k neighbors. We anticipate that a small number of k neighbors is enough to learn the data characteristics and to predict accurately, while separating the influence that the inconsistent and correct data affect each other in the learning process. We examine this capacity of kNN regression with a small number of k neighbors and compare it to another machine learning method, Generalized Regression Neural Network (GRNN). GRNN uses all data instances in order to predict new data, and therefore, the inconsistent and correct data affect each other in the learning process.

Regarding the prediction performance measure, one can use mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), or mean percentage errors (MPE). These measures indicate the errors between the predicted and target values. We find these traditional measures unsuitable for the *proper-inconsistency* problem for several reasons. For instance, from the aforementioned example, the error between the target, 0mm, and prediction, 2.5mm is 2.5mm. Suppose we achieve an error of 5mm from another machine learning method. Considering the context of this problem, both the 2.5mm and 5mm errors may be unacceptable since the difference is too high. However, to some extent, the 5mm error may be useful in identifying the location of inconsistency explained above, even though the error is numerically higher. To address this problem, we propose a new prediction performance measure called *mean acceptable error* (MACE) to measure the performance of prediction models constructed with the presence of *proper-inconsistency*.

For the rest of the paper, we briefly review related research in Chapter 2. In Chapter 3, we discuss the MACE measure, kNN regression, and GRNN algorithms. Then, we apply these methods and provide results on the weldability prediction problem in Chapter 4. We conclude the paper with summary and future works in Chapter 5.

2. LITERATURE REVIEW

Ouafi *et al.* [1] develop an on-line quality assessment system for the resistant spot welding process. The system is based on Neural Network and is able to predict the welding quality, such as nugget width and penetration, when different welding parameters are used. Kim *et al*. [11] apply Neural Network to arc welding parameter selection problems. The system developed is able to determine welding parameters and to avoid inappropriate welding design. Nagesh and Datta [13] use a Neural Network model to predict the weldability in the metal-arc welding process. Welding parameters considered include bead geometry and penetration, and they show that the Neural Network model predicts the weldability more precisely. Pal *et al*. [9] develop a neural network-based system to monitor the weld joint strength in pulsed metal inert gas welding. The system showed good performance in predicting the weld joint strength with relatively few errors. Gohsal and Chaki [4] present a hybrid approach that combines Neural Network and Bayesian regularization. The method is applied to predict the penetration depth performed in the hybrid laser bean welding. Raghavendra *et al*. [7] combine a Neural Network model and ant colony optimization algorithm in order to predict the weld joint strength for the pulsed metal inert gas welding.

Inconsistent data have typically the same values for the input features, but have different output values. They are usually considered as noise in the literature. Gamberger *et al*. [14] define inconsistency as a type of measurement error. Many studies have been conducted to reduce noise using anomaly detection methods. A few recent surveys on anomaly detection are available in [6, 10]. Instance selection methods aim to select relevant instances resulting in eliminating irrelevant, redundant, and noise in data. Recent surveys on different methods and analysis on real-world datasets are also available in [3, 5, 12].

k-Nearest Neighbor (kNN) is a machine learning scheme that can be applied to classification or regression problems. It is known for its simplicity while providing high efficiency and effectiveness. Many researchers employ kNN algorithm in different applications. Detailed information on kNN regression is available in [19]. Another related research area focuses on developing efficient data structures in order to calculate the distances between data instances, and a recent survey on these research works is available in [16]. Generalized Regression Neural Network (GRNN) is an instance-base learning algorithm, which was first introduced in [15]. It has been used in many applications where a function approximation is required. In [2], GRNN is applied to the feature selection problem. They measure the performance of different subsets of features using GRNN. Şenkal [17] develop a solar radiation prediction system using GRNN. Input features to the GRNN model include latitude, longitude, and a few coefficients achieved by satellites. Li *et al*. [18] present a GRNN based approach along with a new optimization algorithm for the parameter for GRNN. The developed approach is applied to predict the annual power load while providing a more tractable computational cost and higher performance compared to other methods.

3. PROPOSED METHODS

3.1. MEAN ACCEPTABLE ERROR (MACE)

The prediction performance is typically measured using MSE, RMSE, MAE, and MPE for machine learning algorithms. For example, MSE is represented as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{1}$$

where, N is the number of data instances and y_i and $\hat{y}_i$ are the target and predicted values for data instance i respectively.

We find these measures unsuitable for the weldability prediction problem for the reasons explained in Chapter 1. Therefore, we define MACE to measure the performance of prediction models as follows:

$$MACE = \frac{1}{N}\sum_{i=1}^{N}\left[\left|y_i - \hat{y}_i\right| \le T\right] \tag{2}$$

where, $[\ ]$ is the Iverson bracket and T is a threshold. The Iverson bracket returns 1, if the statement inside the bracket is true otherwise 0. T can be specified such that the prediction values are reasonably acceptable compared to its target value depending on the context of the problem. In other words, MACE measures the performance of prediction models as in the *number of acceptable predictions*. In this study, we specify a threshold T as follows:

$$T = 0.1 \times \frac{1}{N}\sum_{i=1}^{N} y_i$$

Essentially, T is used to define predictions that are in the 10 percent range of the mean target value as acceptable and therefore correctly predicted.

3.2. K NEAREST NEIGHBOR REGRESSION AND GENERALIZED REGRESSION NEURAL NETWORK

In this section, we introduce k Nearest Neighbor (kNN) regression and Generalized Regression Neural Network (GRNN). kNN regression predicts a new data instance based on the similarities between the new data instance and those in the training data set. The similarities can be calculated using Euclidean or any other distance metrics such as Cosine distance. Once the distances are measured, k nearest data instances and their target values determine the new data instance's prediction value. The Euclidean distance can be calculated as follows:

$$D^2(X, X_j) = D_j^2 = (X - X_i)^T \cdot (X - X_i) \tag{3}$$

where, $X = (x_1, x_2, ..., x_p)$ and $X_j = (x_{j1}, x_{j2}, ..., x_{jp})$ are the new data instance and jth training data instance respectively.

kNN regression can be performed using the following equation:

$$\hat{Y}(X) = \sum_{X' \subset N(X)} \frac{Y(X')e^{(-D_j^2/2\sigma^2)}}{\sum_{X' \subset N(X)} e^{(-D_j^2/2\sigma^2)}} \tag{4}$$

where, $\hat{Y}(X) = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_l)$ is the predicted value of the new data instance X, n is the number of training data instances, X' is the k nearest training data instances that are closest to the new data instance to be predicted X, and e is the exponential function. Each training data instance is used with a weight function, specifically $e^{(-D_j^2/2\sigma^2)}$ to predict the new input data. Each of these distances contributes to the predicted value. Note that the weight function term $e^{(-D_j^2/2\sigma^2)}$ increases when the distance D_j^2 is close and decreases as the distance increases. Therefore, the

response value $Y(X)_j = (y_{j1}, y_{j2}, ..., y_{jl})$ of training data instances that are closer to the new data instance X has more contributions to the final prediction value $\hat{Y}(X)$. The smaller σ becomes, the more each training data contributes to the prediction.

GRNN is an instance-based learning algorithm. The computational cost in training and predicting is generally tractable, compared to those that are not instance-based learning algorithms (e.g., Neural Network with back-propagation algorithm). Another advantage is that there is only one parameter called smoothing parameter, σ, which makes it easy to select the optimal parameter. Some other advantages avoiding a local minima and over-fitting and robustness to outliers are reported in [2]. Using GRNN, predictions can be performed by first calculating the distance between the new data instance and training data instances using the same equations defined above for kNN regression.

Similar to kNN regression, GRNN predicts a new data instance as follows:

$$\hat{Y}(X) = \sum_{j=1}^{n} \frac{Y(X)_j e^{(-D_j^2/2\sigma^2)}}{\sum_{j=1}^{n} e^{(-D_j^2/2\sigma^2)}} \tag{5}$$

Note that GRNN uses all the training data instances to predict $\hat{Y}(X)$ instead of using k nearest training data instances.

Most machine learning algorithms attempt to minimize the overall errors between the predicted and target values. That is the correct approach, in general, since inconsistency is assumed as noise and thus eliminated before applying any machine learning methods. However, in the case of *proper-inconsistency*, these *properly-inconsistent* data should not be affected by the correct data instances so that they can be further examined. On the contrary, the correct data should not be affected by these *properly-inconsistent* data. With that notion, we attempt to segregate the learning mechanisms for the *properly-inconsistent* data from the correct ones and vice versa using kNN regression with a small number of k. In other words, the *properly-inconsistent* data should learn mostly from themselves, and similarly, the correct ones should do the same by not affecting each other. In order to validate the performance of kNN regression with a small k, we compare its performance measured by MACE to the same of GRNN's performance. This is to see whether using a small k nearest neighbors performs well on predicting both the correct and *properly-inconsistent* data. We present and discuss the results in the next section.

4. RESULTS

In our weldability prediction data set, there are 1,280 data instances. Each data instance has its corresponding 16 input feature values that specify different welding parameters and one output feature value, which is nugget width. These 16 input values consist of material characteristics, welding force, welding current, and other relevant features. We use kNN regression with a number of small k values to construct prediction models and we compare the results with those achieved by GRNN.

We perform 10-fold cross validation to construct the models. The entire data set is randomly shuffled and split into ten folds. The data instances in the first fold are kept for testing the model constructed from the rest of data instances that belong to the nine other folds. The predictions are calculated for the data instances in the first fold. Then, the second fold is held out for testing and the rest is used to train another model. This process is repeated until we obtain all the predictions for the entire data set.

Figure 1 shows how the same welding parameters can result in inconsistent welding quality. We group data instances that have the same welding parameters, which result in 262 different groups and then we plot them on the x-axis. Their nugget width values are plotted on the y-axis. Groups 1 through 76 have no inconsistency as their nugget width values are all zeros. Groups 77 through 103 have a significant inconsistency and groups 104 through 262 have both a slight inconsistency and variations. We name these three types of groups as Group A, Group B, and Group C respectively. Each group in Group A and Group B consists of about 2 to 5 data instances. Whereas in Group C, the number of data instances are about 10 to 20 in most of the groups and majority of data, which is about 80% of the entire dataset fall into this group.

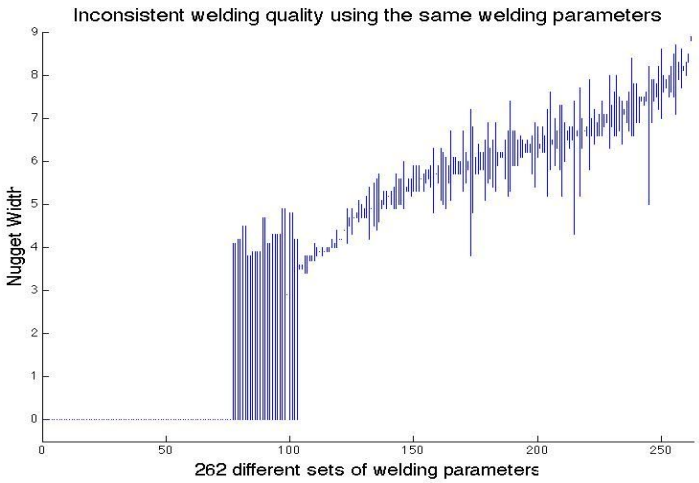

Figure 1. Inconsistent welding quality.

In Table 1, we summarize the experiments conducted using the kNN regression and GRNN. Since we employ 10-fold cross validation by randomly choosing data instances, we replicate 50 times and, for each replication, the prediction performance is measured using the MACE measure we presented in Chapter 3. The threshold was $T = 0.456$ in our experiment which is the mean nugget width calculated from the entire data set.

Table 1. Results of kNN regression and GRNN (minimum, maximum, mean, and standard deviation of MACE).

		$\sigma = 0.01$				$\sigma = 0.05$				$\sigma = 0.15$			
		Group A	Group B	Group C	Total	Group A	Group B	Group C	Total	Group A	Group B	Group C	Total
1NN	min	0.974	0.3465	0.6136	0.5922	0.974	0.3465	0.6148	0.5922	0.9481	0.3343	0.6098	0.5906
	max	0.9935	0.3708	0.6399	0.6117	0.9935	0.3708	0.6437	0.6125	0.987	0.3647	0.6386	0.6086
	mean	0.9862	0.3597	0.6272	0.6016	0.9864	0.3583	0.6267	0.601	0.9855	0.3573	0.6272	0.6009
	sd	0.0039	0.006	0.0061	0.0044	0.0033	0.0062	0.0061	0.0041	0.0062	0.0059	0.0066	0.0043
3NN	min	0.961	0.2401	0.6713	0.6039	0.9091	0.2158	0.6612	0.5906	0.8766	0.234	0.655	0.5844
	max	0.987	0.304	0.6989	0.6266	0.9416	0.3009	0.6964	0.6203	0.9221	0.304	0.6888	0.6094
	mean	0.9808	0.2758	0.6866	0.6164	0.9318	0.2731	0.6821	0.607	0.9019	0.276	0.6707	0.5971
	sd	0.008	0.0147	0.0067	0.0054	0.0075	0.0191	0.0077	0.0069	0.0098	0.0166	0.0067	0.0057
9NN	min	0.9481	0	0.7077	0.557	0.8636	0	0.7001	0.5453	0.8312	0	0.6775	0.5227
	max	0.974	0.0213	0.734	0.5781	0.9156	0.0213	0.7252	0.5617	0.8506	0.0061	0.7014	0.5383
	mean	0.9666	0.0095	0.7205	0.5673	0.893	0.007	0.7135	0.5535	0.8401	0.0017	0.6889	0.5305
	sd	0.0097	0.0049	0.0068	0.0046	0.0111	0.0047	0.0068	0.0043	0.0055	0.0019	0.0056	0.0038
21NN	min	0.9286	0.003	0.6964	0.55	0.8701	0	0.6813	0.5352	0.8052	0	0.66	0.5094
	max	0.974	0.0213	0.7365	0.575	0.9091	0.0182	0.7164	0.5563	0.8247	0.0061	0.6863	0.5266
	mean	0.9669	0.0097	0.7124	0.5624	0.8936	0.0053	0.7025	0.5463	0.8205	0.0005	0.6737	0.5183
	sd	0.0093	0.0047	0.0079	0.0052	0.0101	0.0041	0.0074	0.0051	0.0039	0.0013	0.0068	0.0042
GRNN	min	0.9221	0	0.6926	0.5484	0.8766	0	0.6888	0.5383	0.7857	0	0.6575	0.5055
	max	0.974	0.0243	0.7265	0.5727	0.9156	0.0152	0.7152	0.5539	0.8052	0.003	0.6863	0.5227
	mean	0.9683	0.0095	0.7118	0.5622	0.8938	0.0048	0.7014	0.5455	0.794	0.0008	0.6715	0.5139
	sd	0.0105	0.0054	0.0077	0.0057	0.0087	0.0036	0.0058	0.0037	0.0044	0.0013	0.0059	0.0037

		$\sigma = 0.5$				$\sigma = 1$			
		Group A	Group B	Group C	Total	Group A	Group B	Group C	Total
1NN	min	0.974	0.3435	0.6136	0.5906	0.974	0.3374	0.611	0.5883
	max	0.9935	0.3769	0.6386	0.6086	0.9935	0.3769	0.6437	0.6125
	mean	0.9865	0.3599	0.626	0.601	0.9869	0.3581	0.6268	0.6011
	sd	0.0029	0.0074	0.0062	0.0043	0.0021	0.0078	0.0063	0.0048
3NN	min	0.8766	0.228	0.6462	0.5773	0.8766	0.234	0.6462	0.5734
	max	0.9156	0.2979	0.6888	0.6062	0.9156	0.2948	0.6801	0.6016
	mean	0.9008	0.2691	0.663	0.5904	0.8982	0.2694	0.6609	0.5888
	sd	0.0091	0.0138	0.0079	0.0063	0.0098	0.017	0.0085	0.0067
9NN	min	0.7987	0.003	0.6211	0.4867	0.7922	0	0.6098	0.4766
	max	0.8377	0.0122	0.655	0.5086	0.8312	0.0091	0.6399	0.4984
	mean	0.8149	0.006	0.6394	0.4977	0.8105	0.0055	0.6258	0.4886
	sd	0.0104	0.0023	0.0063	0.0041	0.0097	0.0025	0.0071	0.0043
21NN	min	0.6558	0	0.5885	0.4477	0.5844	0	0.5521	0.4234
	max	0.7013	0.003	0.6161	0.4672	0.6818	0.003	0.5809	0.4398
	mean	0.679	0.0001	0.6007	0.4558	0.6325	0.0001	0.5678	0.4297
	sd	0.0103	0.0006	0.0061	0.0042	0.0232	0.0006	0.0062	0.0041
GRNN	min	0.4351	0	0.5822	0.4164	0.1623	0	0.5207	0.3461
	max	0.4675	0	0.6048	0.4297	0.2078	0.003	0.5433	0.3617
	mean	0.4478	0	0.5938	0.4236	0.1851	0.0001	0.5334	0.3544
	sd	0.0081	0	0.0046	0.0029	0.01	0.0004	0.0048	0.0034

The results show that using a small number of k neighbors provides a better MACE performance than that of GRNN in Group A and B. For instance, we see that from Table 1, when compared to GRNN with $\sigma = 0.01$, kNN regression with $K = 1$ has higher performance for every group. Similarly, kNN with $K = 3$ provides higher MACE performance for Group A and B resulting in a better performance throughout the entire data set. When less K training data instances are used, we also observe that kNN is generally robust to the parameter σ. It doesn't seem that there is a strong correlation between K and the standard deviation even though the standard deviation increases in many cases as K increases. In addition, for Group B where the most significant inconsistency exists, we observe that a smaller number of k neighbors achieves a better MACE performance as kNN with $K = 1$ provides MACE performance is about 35.97% acceptable in terms of the mean performance calculated from the 50 replications. Whereas using GRNN, the inconsistent data could not be predicted in the acceptable range. This is because kNN, with a small number of k neighbors, predicts the *properly-inconsistent* data mostly based on the similar ones from the training dataset. As we increase K, we can see the prediction performance becomes closer to that of GRNN where it uses every data instance in the training dataset to predict a new data instance.

5. CONCLUSION

In this paper, we investigate the inconsistent quality problem in resistance spot welding (RSW). We define this inconsistency as *proper-inconsistency* since they capture the nature of the RSW process. We claim that they should not be treated as noise. Therefore, our attempt was to identify a solution for machine learning algorithms to perform better in the presence of *proper-inconsistency* in the dataset. We proposed a prediction performance measure, called mean acceptable error (MACE), which was used to measure the prediction performance of kNN regression and GRNN. We showed using a smaller number of k neighbors provides a better MACE performance in predicting *properly-inconsistent* data.

We are interested in further examining the possibility of adjusting the number k neighbors for different types of data characteristics. We also consider other applications than RSW where there is a significant inconsistency in data that should not be treated as noise. Some other future works focus on examining other machine learning algorithms and comparing their behaviors in learning from similar inconsistent data sets by using different types of performance measures compared to MACE. In addition, we will study how our approach can be used to support in making decisions (e.g., quality monitoring and material selection) for intelligent manufacturing.

ACKNOWLEDGEMENTS

This research is partially supported by the NSF I/UCRC for e-Design and Ford Motor Company.

REFERENCES

[1] A. E. Ouafi, R. Bélanger, and J. Méthot: "Artificial neural network-based resistance spot welding quality assessment system", Revue de Métallurgie, Vol. 108, No. 6, pp.343–355, 2011.

[2] I. A. Gheyas and L. S. Smith: "Feature subset selection in large dimensionality domains", Pattern Recognition, Vol. 43, No. 1, pp.5–13, 2010.

[3] H. Liu, Instance selection and construction for data mining, Springer-Verlag, 2010.

[4] S. Ghosal and S. Chaki: "Estimation and optimization of depth of penetration in hybrid CO2 LASER-MIG welding using ANN-optimization hybrid model", The International Journal of Advanced Manufacturing Technology, Vol. 47, No. 9, pp.1149–1157, 2010.

[5] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and J. Kittler: "A review of instance selection methods", Artificial Intelligence Review, Vol. 34, No. 2, pp.133–143, 2010.

[6] V. Chandola, A. Banerjee, and V. Kumar: "Anomaly detection: A survey", ACM Computing Surveys (CSUR), Vol. 41, No. 3, pp.1–58, 2009.

[7] N. Raghavendra, R. Koranne, S. Pal, S. K. Pal, and A. K. Samantaray: "Joint strength prediction in a pulsed MIG welding process using hybrid neuro ant colony-optimized model", The International Journal of Advanced Manufacturing Technology, Vol. 41, No. 7, pp.694–705, 2009.

[8] G. Xu, J. Wen, C. Wang, and X. Zhang, "Quality monitoring for resistance spot welding using dynamic signals", IEEE International Conference on Mechatronics and Automation, pp.2495–2499, 2009.

[9] S. Pal, S. K. Pal, and A. K. Samantaray: "Artificial neural network modeling of weld joint strength prediction of a pulsed metal inert gas welding process using arc signals", Journal of materials processing technology, Vol. 202, No. 1, pp.464–474, 2008.

[10] V. J. Hodge and J. Austin: "A survey of outlier detection methodologies", Artificial Intelligence Review, Vol. 22, No. 2, pp.85–126, 2004.

[11] I.-S. Kim, Y. Jeong, C. Lee, and P. Yarlagadda: "Prediction of welding parameters for pipeline welding using an intelligent system", The International Journal of Advanced Manufacturing Technology, Vol. 22, No. 9, pp.713–719, 2003.

[12] H. Liu and H. Motoda: "On issues of instance selection", Data mining and knowledge discovery, Vol. 6, No. 2, pp.115–130, 2002.

[13] D. Nagesh and G. Datta: "Prediction of weld bead geometry and penetration in shielded metal-arc welding using artificial neural networks", Journal of materials processing technology, Vol. 123, No. 2, pp.303–312, 2002.

[14] D. Gamberger, N. Lavrač, and S. Džeroski, Noise elimination in inductive concept learning: A case study in medical diagnosis, Algorithmic Learning Theory, Springer Berlin Heidelberg, pp.199–212, 1996.

[15] D. F. Specht: "A general regression neural network", IEEE transactions on neural networks, Vol. 2, No. 6, pp.568–576, 1991.

[16] N. Bhatia and Vandana: "Survey of nearest neighbor techniques", International Journal of Computer Science and Information Security, Vol. 8, No. 2, pp.302-305, 2010.

[17] O. Şenkal: "Modeling of solar radiation using remote sensing and artificial neural network in Turkey", Energy, Vol. 35, No. 12, pp.4795–4801, 2010.

[18] H. Z. Li, S. Guo, C. J. Li, and J. Q. Sun: "A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm", Knowledge-Based Systems, Vol. 37, pp.378–387, 2012.

[19] C. G. Atkenson, A. W. Moore, and S. Schaal: "Locally weighted learning for control", Artificial Intelligence Review, Vol. 11, pp.75–113, 1997.